IN THE CLAIMS:

Claims 1-12 canceled.

13. (Previously presented) A method for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system comprising:

determining the number of devices being serviced via the ports,

identifying at least one assembly for hot swapping,

copying the contents of cache memories associated with the at least one identified assembly,

setting criteria for transactions at the port with respect to the number of devices, and

with respect to the numbers of devices at the ports, assigning resources to the ports.

14. (Currently amended) The method as defined in claim 13 wherein assigning ~~system~~ resources to the ports comprises at least one of assigning control registers to the ports, assigning direct memory access engines to the ports, assigning cache memory to the ports and assigning priorities among the transactions at the ports.

15. (Previously presented) A system for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system, the system comprising:

means for determining the number of devices being serviced via a port,

at least one assembly identified for hot swapping,

means for copying the contents of cache memories associated with the at least one identified assembly,

means for setting criteria for transactions at the port with respect to the number of devices, and

means, responsive to the criteria, for assigning resources to the ports.

16. (Previously presented) The system as defined in claim 15 wherein the resources assigned to the ports comprises at least one of

direct memory access (DMA) engines,

cache memory, and

means for assigning priorities among the transactions at the ports.

17 (Previously presented) The method as defined in claim 13 further comprising determining the number and types of transactions anticipated at the ports, wherein the assignment of resources is further with respect to the numbers and types of transactions at the ports.

18. (Previously presented) The method as defined in claim 13 wherein the at least one identified assembly has a memory system, and the method further comprises copying the states and status of the memory systems associated with at least one identified assembly.

19. (Previously presented) The system as defined in claim 15 further comprising means for determining the number and types of transactions anticipated at the ports, wherein the criteria further accounts for the anticipated number and types of transactions.

20. (Previously presented) The system as defined in claim 15 wherein the at least one identified assembly has a memory system, and the system further comprises means for copying the states and status of the memory systems associated with the at least one identified assembly.

21. (Previously presented) A method for programmably allocating resources for processing Input/Output (I/O) transactions at a plurality of I/O ports of an I/O bridge, the method comprising:

identifying the number of I/O devices being serviced by at least one I/O port;

setting criteria for the transactions at the at least one I/O port with respect to the number of I/O devices being serviced by the port; and

assigning the resources to the at least one I/O port in response to the criteria.

22. (Previously presented) The method of claim 21 wherein the assigning comprises assigning a plurality of direct memory access (DMA) engines for use in processing I/O transactions.

23. (Previously presented) The method of claim 22 wherein assigning comprises apportioning a selected number of DMA engines to process a given transaction at a particular I/O port.

24. (Previously presented) The method of claim 22 wherein assigning comprises apportioning at least one DMA engine to process at least one transaction at a port.

25. (Previously presented) The method of claim 22 wherein assigning comprises apportioning one DMA engine to process a given transaction at a port identified as servicing multiple I/O devices.

26. (Previously presented) The method of claim 21 wherein assigning comprises assigning at least one miss address file (MAF) value for processing I/O transactions.

27. (Previously presented) The method of claim 21 wherein assigning comprises assigning a plurality of miss address file (MAF) values for processing I/O transactions.

28. (Previously presented) The method of claim 27 further comprising reducing the assigned number of MAF values.

29. (Previously presented) The method of claim 21 wherein

the I/O bridge is configured to utilize a plurality of virtual channels to communicate with at least one processors of a multiprocessor computer system, and

the resources include flow control credits associated with each of the plurality of virtual channels.

30. (Previously presented) The method of claim 29 wherein assigning comprises setting the number of flow control credits associated with each virtual channel.

31. (Previously presented) The method of claim 21 wherein

the I/O bridge comprises at least one control register, the at least one control register having a plurality of fields, and at least one field of the control register being associated with a corresponding resource, and

the method further comprises writing to a selected field of the at least one control register so as to modify the assignment of resources.

32. (Previously presented) An Input/Output (I/O) bridge for use in a computer system having a plurality of processors, the I/O bridge comprising:

a plurality of I/O ports, each I/O port configured to communicate with at least one I/O device that generates or receives transactions;

resources for use in servicing the transactions of the I/O devices; and

programmable logic configured and arranged to assign the resources among the I/O ports in response to the number of I/O devices with which the I/O ports are communicating.

33. (Previously presented) The I/O bridge of claim 32 wherein

the resources comprise at least one direct memory access (DMA) engine configured to process the transactions, and

the programmable logic apportions the at least one of DMA engine to process at least one transaction at a given I/O port in response to the number of I/O devices coupled to the given I/O port.

34. (Previously presented) The I/O bridge of claim 32 wherein

the resources include a plurality of miss address file (MAF) values for use in requesting information from the computer system, and

the programmable logic sets the number of available MAF values.

35. (Previously presented) The I/O bridge of claim 32 wherein

the I/O bridge communicates with the computer system through a plurality of virtual channels,

the resources include a plurality of flow control credits associated with the virtual channels, and

the programmable logic assigns a number of flow control credits to each virtual channel.

36. (Previously presented) the I/O bridge of claim 35 wherein the virtual channels comprise a Request channel, a Read I/O channel, and a Write I/O channel.

37. (Previously presented) The I/O bridge of claim 33 further comprising at least one cache for storing information, wherein, to hot-swap an assembly of the computer system, the programmable logic is configured to

disable the at least one DMA engine, and

flush the information from the at least one cache.

38. (Previously presented) The I/O bridge of claim 37 wherein the at least one cache is one of a write cache, a read cache and a translation look-aside buffer (TLB).

39. (Previously presented) The I/O bridge of claim 37 wherein the assembly is a processor.

40. (Previously presented) The I/O bridge of claim 33 wherein

the programmable logic comprises at least one control register associated with each I/O port, and

the at least one control register has a first field for apportioning the at least one DMA engine.

41. (Previously presented) The I/O bridge of claim 32 wherein the programmable logic re-assigns resources among the I/O ports dynamically while the I/O bridge continues to operate.